UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/001,478	11/01/2001	Craig Nemecek	CYPR-CD01213M	6435

45545 7590 10/20/2008
CYPRESS C/O MURABITO, HAO & BARNES LLP
TWO NORTH MARKET STREET
THIRD FLOOR
SAN JOSE, CA 95113

EXAMINER	
PROCTOR, JASON SCOTT	

ART UNIT	PAPER NUMBER
2123	

MAIL DATE	DELIVERY MODE
10/20/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/001,478	NEMECEK ET AL.
	Examiner	Art Unit
	JASON PROCTOR	2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 September 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-10,12 and 14-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-10,12 and 14-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claims 1, 2, 4-10, 12, and 14-23 were rejected in the Final Office Action entered on 4 June 2008.

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 4 September 2008 has been entered.

The 4 September 2008 submission has amended claims 1, 10, and 12. Claims 1, 2, 4-10, 12, and 14-23 are pending in this application.

Claims 1, 2, 4-10, 12, and 14-23 are rejected.

Priority

1. This Application contains a claim for the benefit of priority to U.S. Provisional Application No. 60/243,708 filed 26 October 2000. The provisional application has been reviewed and priority is denied, because the provisional application does not appear to enable the claimed invention as required under 35 U.S.C. Section 112, first paragraph. See 35 U.S.C. § 119(e)(1).

For example, the provisional application contains a set of 'powerpoint-style' drawings and datasheets describing desired features for a microcontroller or a 'system-on-chip,' but this material does not appear to contain either the text description or the drawings found in the

Application. In particular, no part of the provisional application appears to disclose the method steps shown in the Application at Fig. 7.

Response to Arguments – 35 USC § 103

2. In response to the previous rejection of claims 1, 2, 4-10, 12, and 14-23 under 35 U.S.C. §103 as being unpatentable over Coker in view of Rosenberg, further in view of Teramoto, Applicants argue primarily that:

> [...] Applicants respectfully submit that Coker fails to teach or suggest the features [of "in the microcontroller, executing a set of boot code to carry out initialization, in the virtual microcontroller, executing a set of timing code to enable a lock-step synchronization"] because it fails to teach or suggest execution of different software as claimed.

The Examiner respectfully traverses this argument as follows.

As deliberately pointed out by the Examiner in the previous Office Action, the Coker reference ***explicitly*** teaches executing different software on a target-ECS (microcontroller) and a shadow system (virtual microcontroller) [*"Thus, rather than receiving input data from an external source and reading the data into complex I/O registers, the shadow system uses the data value and relative time of input events from the target-ECS and writes the value directly to its RAM using its internally generated location."* (Coker, column 2, line 63 – column 3, line 1)]. It would have been well known to persons of ordinary skill in the art long before the time of Applicants' invention that ***instructions for receiving input data from an external source and reading the data into complex I/O registers*** are **different software instructions** than ***instructions to read or write data to RAM***. In particular, **instructions to perform input/output (I/O) operations are different than instructions to store data in memory**. Therefore, Coker clearly and deliberately teaches that the target-ECS (microcontroller) and the shadow system (virtual microcontroller) execute different software.

3. Applicants further argue that:

> Coker appears to [teach that the terms "software," "code" or "instructions" are used interchangeably according to their dictionary definitions] where Claim 12 [i.e., Claim 12 of the Coker patent] recites that "... said shadow system duplicates the software instruction of said target system without interfering with the operation of said target system." Therefore, Applicants assert that the software executed by the shadow system in the reference is not similar to the one executed by the target-ECS but is duplicated from the software of the target-ECS.

The Examiner respectfully traverses this argument as follows.

Coker teaches **how the target-ECS and shadow system software are different** [*"Thus, rather than receiving input data from an external source and reading the data into complex I/O registers, the shadow system uses the data value and relative time of input events from the target-ECS and writes the value directly to its RAM using its internally generated location."* (Coker, column 2, line 63 – column 3, line 1)] and Coker presents claim language that reflects **how the target-ECS and shadow system software are different** [*"The system of claim 1 wherein said shadow system duplicates the software instruction of said target system without interfering with the operation of said target system."* (Coker, claim 12)]. Applicants' argument completely overlooks the claim language **"without interfering with the operation of said target system"** especially in light of Coker's explicit disclosure that the shadow system operates differently than the target-ECS with respect to receiving input data **from an external source** and reading the data **into complex I/O registers**.

If Coker's shadow system executed exactly the same instructions as the target-ECS as Applicants allege, Coker would not disclose that the shadow system executes instructions

differently with respect to complex I/O registers and Coker would not claim that the shadow system "duplicates" the software instruction of said target system with the additional claim limitation that the "duplicating" happens "without interfering with the operation of said target system".

4. Applicants further argue that:

> Unlike Coker which teaches executing the same software or allegedly similar but different instructions to cater to the two architecturally different processors, the microcontroller recited in Claim 1 performs initialization of the microcontroller by running a set of boot code, whereas the virtual microcontroller performs synchronization with the microcontroller by running a timing code.
>
> That is, Coker teaches executing the same or similar software so that the shadow system can shadow the target-ECS, whereas the virtual microcontroller according to Claim 1 does not shadow the microcontroller during its initialization since the set of timing code is executed in the virtual microcontroller while the set of boot code is executed in the microcontroller. Since Coker fails to teach or suggest the claimed features of "in the microcontroller, executing a set of boot code to carry out initialization, in the virtual microcontroller, executing a set of timing code to enable a lock-step synchronization," Applicants respectfully submit that Claim 1 overcomes this reference [...]

The Examiner respectfully traverses this argument as follows.

The rejection of claim 1 under 35 U.S.C. § 103 involves a combination of three references. Applicants must establish that the claimed invention is non-obvious over the combination of these three references. Here Applicants argue that Coker does not teach the claimed "dummy code". Teramoto teaches the claimed "dummy code" [*"The present invention relates to an instruction execution control method and information processing apparatus for monitoring information about the completion of synchronization between processors, and selectively causing a specific instruction of the subsequent group of instructions to wait until the completion of synchronization is indicated when the processors are operated in synchronism with each other for synchronous execution of their respective processes in a computer system including a plurality of processors."* (column 1, lines 7-16); *"When the instruction, which was* *read out, is a Wait instruction, the instruction analyzer 102 issues a Wait instruction to the Wait instruction controller 100 through the interface signal 105. The Wait instruction controller 100 executes the Wait instruction which controls the instruction execution sequence according to the present invention."* (column 5, lines 50-60)].

Applicants submit similar arguments for claims 2 and 4-9, 10, and 12. These arguments have been traversed above.

5. In response to the rejection of claim 22, Applicants argue primarily that:

> Sine Coker teaches a shadow system executing the same software as the target-ECS from system start-up or reset, it fails to teach or suggest that the proprietary information contained in the set of boot code of the microcontroller (which allegedly corresponds to the target-ECS) is inaccessible to the virtual microcontroller (which allegedly corresponds to the shadow system). Since Coker fails to teach the claimed limitations of "the set of boot code comprises proprietary information, wherein the proprietary information comprises serial numbers, passwords, and algorithms," Applicants respectfully assert that Claim 22 overcomes the rejections of record [...]

The Examiner respectfully traverses this argument as follows.

Claim 22 has no limitations whatsoever directed to "inaccessible" proprietary information. Further, this language describes a **negative limitation** which is inherently broad. Coker's disclosure of two processors executing similar (or, for arguments' sake, perfectly identical) instructions **in no way equates to one processor <u>accessing</u> information stored on the other processor**. In fact, Coker discloses no means, methods, or mechanisms by which the shadow system (virtual microcontroller) could access information stored on the target-ECS (microcontroller). Because there is no disclosure in the prior art that it is possible for the shadow system to access information stored on the target-ECS, and **there is no reasonable basis for concluding that the shadow system actually performs a step of "accessing" information**

stored on the target-ECS, the Examiner rightfully concludes that information stored on the target-ECS is "inaccessible" to the shadow system.

Applicants' arguments have been fully considered but have been found unpersuasive.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. § 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. § 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. § 103(c) and potential 35 U.S.C. § 102(e), (f) or (g) prior art under 35 U.S.C. § 103(a).

6. Claims 1-2, 4-10, 12, 14-20, and 23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over US Patent No. 5,371,878 to Coker in view of "How Debuggers Work" by Jonathan B. Rosenberg (Rosenberg), further in view of US Patent No. 5,968,135 to Teramoto et al.

Regarding claim 1, Coker teaches a method comprising:

In a microcontroller, executing a set of boot code to carry out initialization ("Embedded Computer System or ECS", see column 1, lines 20-23) [*"[T]his invention uses hardware and software which can 'shadow' the execution of a target-ECS in real time operation"* (column 2, lines 34-40); *"A shadow system of this invention executes the same software as the target-ECS from system start-up or reset."* (column 2, lines 56-58); Applicants' remarks state that: "Booting is a process that starts a system (e.g., operating system) and substantially initialize the system when a user turns on a computing system or a system with a processor." (Applicants' remarks, 6 February 2007, page 15)];

In a virtual microcontroller ("shadow system"), executing a set of timing code to enable the lock-step synchronization with the microcontroller [*"A shadow system of this invention executes the same software as the target-ECS from system start-up or reset."* (column 2, lines 56-58); *"The shadow system and the target-ECS function exactly the same except that the shadow system receives data slightly delayed because of a data buffer between the target-ECS and the shadow system."* (column 3, lines 13-16); *"[I]n terms of relative time, the execution* *state of the shadow system 28 at the time when any given instruction is executed will directly correspond to the execution state of the target-ECS when the same instruction was executed.."* (column 8, lines 44-49)]

and wherein the set of timing code is different from said set of boot code, [*"Thus, rather than receiving input data from an external source and reading the data into complex I/O registers, the shadow system uses the data value and relative time of input events from the target-ECS and writes the value directly to its RAM using its internally generated location."* (column 2, line 63 – column 3, line 1)];

and wherein the set of boot code is stored within the microcontroller and the set of boot code is inaccessible to the virtual microcontroller [Interface means 19 connecting the target-ECS and shadow system is used by Coker to transmit I/O data and does not appear to contain any suggestion that instruction code or boot code is transmitted via interface means. See (column 4, lines 9-18)].

Although Coker teaches "a computer system and method for debugging, verifying and developing an embedded computer system" (column 1, lines 9-11), Coker does not expressly teach old and well-known tools and techniques of debugging. As a result, Coker teaches a method of debugging, but does not expressly teach simultaneously halting both the microcontroller and the virtual microcontroller.

Rosenberg expressly teaches halting multiple processors operating in lockstep [*"A significant issue for debuggers on multiprocessor systems is how or whether other processors stop when a fault occurs in one. On SIMD architectures, by definition, all processors operate in lock step and so this is guaranteed."* (page 45, first full paragraph)].

Rosenberg and Coker are analogous art because both are drawn to debugging.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Rosenberg and Coker because Rosenberg provides teachings specifically directed to debugging tools, which are suggested by Coker, but "are very difficult tools to build robustly because they depend heavily on relatively weak portions of operating systems and because they tend to stress the underlying operating system's capabilities." (Rosenberg, page 1, first paragraph). That is, Rosenberg provides the teachings that provide helpful guidance in making and using the "critical tools for the development of software" (Rosenberg, page 1, first paragraph) that are suggested by Coker (title, abstract). The combination could be achieved as expressly taught by Rosenberg, by halting the target-ECS and the shadow system "simultaneously" because Rosenberg explicitly teaches this behavior in architectures, like the one claimed, that operate in lock step. (*Id.*)

Coker in view of Rosenberg does not expressly teach that the timing code is a dummy code timed to take the same number of clock cycles as the microcontroller uses to execute the set of boot code. Applicants have defined "dummy code" as exemplified in the remarks submitted on 31 October 2007.

Teramoto teaches timing code that is a dummy code timed to take the same number of clock cycles as a second processor executing code [*"The present invention relates to an instruction execution control method and information processing apparatus for monitoring information about the completion of synchronization between processors, and selectively causing a specific instruction of the subsequent group of instructions to wait until the completion of synchronization is indicated when the processors are operated in synchronism with each other* *for synchronous execution of their respective processes in a computer system including a plurality of processors."* (column 1, lines 7-16); *"When the instruction, which was read out, is a Wait instruction, the instruction analyzer 102 issues a Wait instruction to the Wait instruction controller 100 through the interface signal 105. The Wait instruction controller 100 executes the Wait instruction which controls the instruction execution sequence according to the present invention."* (column 5, lines 50-60)].

Teramoto and Rosenberg in view of Coker are analogous art because both are drawn to operating processor synchronously.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Teramoto with Rosenberg in view of Coker as expressly motivated by Teramoto to maintain high execution speed [*"An object of the present invention is to provide an instruction execution control method and an information processing apparatus for enabling synchronized operations to be performed at high speed among a plurality of processors sharing a main storage."* (column 2, lines 40-44)].

Therefore it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Teramoto with Rosenberg in view of Coker to achieve the invention specified in claim 1.

Regarding claim 2, Coker teaches copying register contents from the microcontroller to corresponding registers in the virtual microcontroller after completion of the simultaneous halting [*"An execution state, including the contents of RAM and I/O registers in the target-ECS and RAM and I/O state memory in the shadow system, can further be defined as including an* *input state vector, output state vector, and internal state vector. Input and output state vectors are defined as the contents of input and output registers respectively in the target-ECS and as the contents of the I/O memory state in the shadow system."* (column 8, lines 50-57); copied from the target-ECS to the shadow system, *"the shadow system receives its input data from the input registers of the target-ECS and stores the input data in its RAM."* (column 2, lines 61-63)].

Also, Rosenberg teaches that *"debuggers provide disassembly views and direct access to hardware registers."* (page 6, first full paragraph).

Regarding claim 4, Rosenberg teaches that after executing code, a processor branches to an instruction line [*"A breakpoint is a special code placed in the executing code stream that, when executed, causes a special trap to occur that the processor and the operating system report to the debugger."* (page 5, second full paragraph); More generally at pages 56-57, "Generic OS-Debugger Interaction Model", etc.].

Regarding claim 5, Rosenberg teaches that prior to executing code, a break is set at an assembly instruction line [*"A breakpoint is a special code placed in the executing code stream that, when executed, causes a special trap to occur that the processor and the operating system report to the debugger."* (page 5, second full paragraph)].

Regarding claim 6, Coker teaches that the boot code comprises protected initialization code that is not accessible to the In-Circuit Emulation system [Coker teaches no means by which the in-circuit emulation system may access the code on the target-ECS. Here Applicants' have presented a negative limitation, which has been interpreted and treated according to MPEP 2173.05(i).].

Claim 7 recites a combination of limitations recited by claims 4 and 5, which are taught by Coker in view of Rosenberg, further in view of Teramoto as shown above.

Claim 8 recites a combination of limitations recited by claims 2, 4, and 5, which are taught by Coker in view of Rosenberg, further in view of Teramoto as shown above.

Regarding claim 9, Rosenberg teaches removing a break at an assembly line [*"Inactive breakpoints are place holders that the user can turn active but currently will not cause execution to stop if reached."* (page 27)].

Claim 10 recites a combination of limitations recited by claims 1, 2, and 4-9. As Coker in view of Rosenberg, further in view of Teramoto teaches these claims, claim 10 is rejected for similar rationale.

Claims 12-20 present a broader recitation of claims 1, 2, and 4-9. These claims are rejected for rationale similar to that given above for claims 1, 2, and 4-9 as being obvious over Coker in view of Rosenberg, further in view of Teramoto.

Regarding claim 23, Coker teaches that at least on portion of the boot code is stored internally in said microcontroller [*"In other words, an ECS normally executes software..."* (column 1, lines 29-39); As noted above, Coker teaches no means by which the in-circuit emulation system may access the code on the target-ECS.].

7. Claim 21 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Coker in view of Rosenberg, further in view of Teramoto as applied to claim 12 above, and further in view of "Emulation of the Sparcle Microprocessor with the MIT Virtual Wires Emulation System" by Matthew Dahl, Jonathan Babb, Russel Tessier, Silvina Hanono, David Hoki, and Anant Agarwal (Dahl) and further in view of "A Reconfigurable Logic Machine for Fast Event-Driven Simulation" by Jerry Bauer, Michael Bershteyn, Ian Kaplan, and Paul Vyedin (Bauer).

Coker in view of Rosenberg, further in view of Teramoto does not expressly teach that the shadow system is implemented on a field programmable gate array (FPGA).

Dahl teaches that it is known in the art to emulate a Sparc microprocessor using an FPGA (abstract).

Bauer teaches that hardware emulation can increase simulation speed by up to 10,000 times (introduction, paragraphs 1-2).

Therefore it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine these teachings and arrive at the decision to implement the shadow system of Coker on an FPGA to realize an enormous increase in simulation speed.

Knowledge that this was possible is provided by Dahl, and motivation to combine the references, to increase simulation speed, is provided by Bauer.

Therefore it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Coker in view of Rosenberg, further in view of Teramoto with Dahl and Bauer to arrive at the invention specified in claim 21.

8. Claim 22 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Coker in view of Rosenberg, further in view of Teramoto as applied to claim 1 above, and further in view of US Patent No. 4,757,534 to Matyas et al.

Regarding claim 22, Coker in view of Rosenberg, further in view of Teramoto does not disclose that the boot code comprises serial numbers and passwords.

Matyas teaches code that comprises serial numbers, passwords, and algorithms [*"In carrying out the computer/smart card protocol, the T output is sent to the smart card together with the parameters P1* ***(password)*** *and P2 (program number | diskette* ***serial number****, where | denotes concatenation) and a third parameter P3. Where the crypto facility uses the DES* ***algorithm****, as shown in FIGS. 8 and 9, to encrypt the file key, P3 is the computer number."* (column 13, lines 1-21)].

Additionally, Coker teaches computer software comprising algorithms [*"A shadow system of this invention executes the same software as the target-ECS from system start-up or reset."* (Coker column 2, lines 56-58)]; Rosenberg teaches computer software comprising algorithms [*"Debuggers are quite complex pieces of software. Their inner workings require a* *suite of sophisticated algorithms and data structures to accomplish their tasks."* (Rosenberg, page 2)]; Teramoto teaches computer software comprising algorithms (Teramoto, FIG. 10).

Matyas and Coker in view of Rosenberg, further in view of Teramoto are analogous art because both are directed to computer software.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Matyas with Coker in view of Rosenberg, further in view of Teramoto to include the passwords, serial numbers, and algorithms in the boot code in order to discourage "the copying and sharing of purchased software program" (here, the boot code) as expressly taught by Matyas (abstract). The combination could be achieved by implementing the smart card cryptographic method taught by Matyas with the target-ECS system taught by Coker.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Matyas with Coker in view of Rosenberg, further in view of Teramoto to arrive at the invention specified in claim 22.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached at (571) 272-3753. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Jason Proctor/
Examiner
Art Unit 2123

jsp